IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

News aggregator web application using Django

Mr Rakesh Kumar Rai,Dr Isha Singh,Ankit Mudia, Karandeep Bisht

Department of Information technology ,HMRITM,Delhi, India.

Abstract : In this modern world we know that there are a lot of news websites and they cover news on various topics, out of which very few are of our interest. A new aggregator can be a tool to save a lot of time and with some modifications and filtrations we can fine tune it to show only news of our interest. A news aggregator is a very useful application to view information within minimum time. We can build a news aggregator application by scrapping some famous websites and serving the scrapped articles via Django on web or in any app. It is a web application which aggregates data from multiple websites. Then shows the articles on desired pages.

News is a very important start of the day. There are well known news sites online. Now, imagine opening several news websites every single day. The time wasted from reading from those websites and information gaining is also essential. It can give leverage over those who don't have it. Now, we have made this task really easy for everyone!!

In a news aggregator, you can read news from various websites. Then the news aggregator collects the article for you. And just after a click or two, you can view news of your choice.

Keywords : HTML, CSS, Javascript, Django, HTL Python.

I. INTRODUCTION

In the modern society that we are living in, we only want things easily for us. We do not want to go out and pick up a news paper every time or even search google for the latest news, that is where out news aggregator web application comes in hand.

It will allow a user to view or read news conveniently without any hassle, all he/she has to do is just open our news website and he will be able to see all the latest news from two very famous and respected news websites and those news will also be fully trusted.

In today's world trusting a news is a difficult thing for us because of the spreading of fake news, so user will not have to worry about the authenticity of the news as news are from trusted platform.

At the same time, RSS provides condensed item information, a favourite for a news aggregator. Even through there are small discrepancies between RSS formats, they are still an effective solution for indexing articles.

A solution that uses web-crawling and Hyper Text Markup Language (HTML) parsing techniques is not adequate due to the fact that each website has its own structure. Also, even small changes to the Document Object Model (DOM) of the websites can have dire consequences to the parsed results. Another well-known method of fetching articles is based on specific APIs that the news platforms provide. An API only based solution would be limited given the fact that not all news outlets provide each other method. Nonetheless, these solutions can supplement the one based on RSS feeds.

The application delivers a couple of unique features as described below:

This system downloads articles from 2 predefined Reduces the time required to regularly visit websites by bringing information into one place. Users can quickly access the feeds and they can then click on items they find interesting.

1. Simplicity in use, and fluidity of the graphical interface. Although some existing news aggregators can offer much more complex interface and more advanced settings, they can be quite difficult to use, especially for a non-technical person.
2. Spam removal – although we can say that most aggregator offer spam and ad filtering for articles, more and more have begun to add their own ads for profit generation. The developed aggregator does not alter the user experience by adding advertisements.
3. Ability to filter and sort content according to user defined criteria.
4. Provides an API that can be used by users to develop their own custom application. The fact is that most aggregators present in today's world does not provide a good interface to the user.
5. In the second section we take a look on the related work and next section describes the system of the application. We present the application interface in Section 4. First, we discuss news aggregators and news websites and it cannot be configured to add more.
6. News articles are in HTML format from news websites using a Python framework namely scrapy. The main goal is to extract different components of the webpage, such as title, the text of news, lead paragraph, publication data, the news author and the associated image. The news aggregator offers the option to organize the feeds into categories. Besides this, it also creates news summaries which decrease the required reading time. Flipboard is another well know application that manages RSS feeds. Flipboard offers the possibility of grouping feeds into categories as well as editing and adjusting them. The European Media Monitor (EMM) is a sizable project implemented by the European Commission's Joint Research Centre. EMM monitors social media from Europe in real-time gathering daily between 80 and 100.000 news articles in as much as 50 languages. In case websites have RSS support, EMM employs them, otherwise is reverts to HTML parsing.

II. ARCHITECTURE

The application is structured into 3 main parts. These are following;

We will study the HTML source code of news sites we want to scrap and build a website scrapper:-

1. First, we'll setup our Django server
2. Then, we'll integrate everything altogether So, let's

start with first step.

Building the website scrapper

Before we start building our news websites scrapper, let us go and get the required packages first and install them on the system. You can install them from command prompt by these commands. This will install the required packages.

```
pip install bs4
pip install requests
```

FIG 1. Installing required packages

We are going to use Times of India and Hindustan Times as our news sources. We'll Get articles from four websites and then scrap and show it into our news aggregator web application.

First, we will scrap one of the famous news company's website, The Times of India... We'll take news from brief section of times of India. We can see that the news heading are in the h2 tag.

Fig 2. TOI's page to be scrapped

So we will take the h2 tag common for the website. Here is how our scrapper will look like.

Then we will show all the news headings from times of India on our page.

Now, let's move to NDTV. We will scrap the headings section of their website. Here we can see that, news is coming in a div with heading four class.

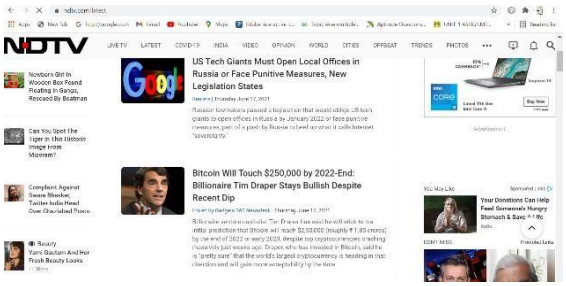

Fig 3. NDTV's page to be scrapped

Now we have the all the news that we have scrapped from other websites on our web app. We can start building our web app. The very first module of this application is the web server which will work on django. This one is processing the HTTP requests received from customers. To develop a Django web application, we need to install Django on our system. After installing Django, we can start building our application. Once we have manage.py file. Django, has convention of keeping everything in separate app, Inside a project. This is the command to create app. I am calling the app News aggregator.

Thus, using a web framework is preferable because the purpose of this paper is to develop a news aggregator rather than a web server. Flask is a popular web framework for Python and it gives users more control over the components they want to use. Flask is also a powerful and flexible solution especially when it comes to developing REST APIs [17]. Because the developed aggregator will provide an API that can be used later to develop a client that will expose all of its features in a simple and easy to use way, Flask is an excellent choice for this task. At the bottom of the REST services, that the application provides, lies the concept of resource [18]. Resources are generally represented by URLs. Customers are sending requests to these URLs using the methods defined by the HTTP protocol, and as a result of these requests, the status of the resources may change or not. In REST services, HTTP request methods have a well-defined role and are typically designed to affect a particular resource in standard ways.

• /get articles - provides a list of articles for a user;

• /get articles/ - provides a list of items from a specific user directory;

• /get feed articles/ - retrieve a list of articles corresponding to a particular news stream;

• /twitter – Provides twitter link on application;

• /news API sources - manage News API sources;

• /news API/ - retrieve articles from News API source.

In this project we have used HTML, CSS along with Bootstrap properties to make the front end of this project along side javascript, the front end of the application was made mainly by HTML and CSS, Bootstrap was used purely for the designing purposes, these technologies were used to make eight to nine pages in total for the display of the web application.

HTML – It is a technology which is used to define the structure of any webpage. It is also used to specify whether your content should be in a paragraph, list, heading, link, image, multimedia player, form, or one of many other available elements or even a new element that you define.

Contents could be structured within a set of paragraphs, bullet points, or using images and data tables.

CSS - Cascading Style Sheets is a style sheet language which is used for describing the styling of a document written in a markup language such as HTML.

Without CSS, every web page would be drab plain text and images that flowed straight down the page. With CSS, you can add colour and background images and change the layout of your page — your web pages can feel like works of art!

CSS handles the overall appearance of the webpage. By using CSS, anyone can control the colour, text, style, fonts, spacing.

Bootstrap – It is a free and an open-source tool of collections which is used for creating responsive websites and web applications. It is known for popular uses with HTML, CSS, and JavaScript framework for developing responsive, mobile and desktop websites. It solves the issues of compatibility. We get Faster and Easier Websites through bootstrap.

a) Development.
b) It creates Platform-independent web pages.

c) It creates Responsive Web-pages.
d) It designed to be responsive to mobile devices too.
e) It is Free! Available on www.getbootstrap.com

It produces less cross-browser bugs. It is a consistent framework supported by all the browsers plus CSS based compatibility fixes. A simple and effective grid system.

News aggregator: In computing terms, news aggregator, is also known as the feed aggregator or news reader or an RSS reader or simply an aggregator and is a client software or a web application that aggregates the web content such as newspapers which are online, blogs, podcasts, and video blogs (vlogs) and all these things are located at a single place.

An aggregator pulls together and allows a user to view news at a single place by using the concept of web scrapping, through web scrapping we can pull the content of almost every website and show it where ever we want or store it as well.

Django - Django is a Python web framework that enables fast development of secure websites. It is a high level frame-work. Built by experienced developers, Django takes care very much of the difficult tasks of web development, so the user can focus on writing the application without worrying for the different tools. Django is free and open source. It also has a vast community.

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django!

Django also helps the developers to avoid many commonly made security mistakes by providing a framework that has been designed to protect the website automatically. It directly stores the password rather than generating the hash key.

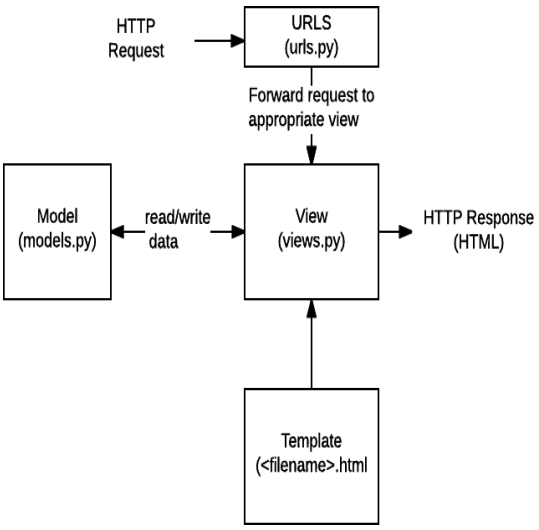

FIG 4. MVT diagram of Django

URL's: It is possible to process requests from single URL via single function and it is more maintainable to write a new view function to handle that resource. The URL mapper is used to match a particular patterns of strings or digits that appear in a URL and pass these to a view function as the data in the views.py file.

View: It is a request handler function, which receives the HTTP requests and returns HTTP responses. It access the data that is needed to satisfy requests via *models*.

Models: Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.

Templates: A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A *view* can dynamically create an HTML page using an HTML template, populating it with data from a *model*. A template can be used to define the structure of any type of file; it doesn't have to be HTML!

Output:

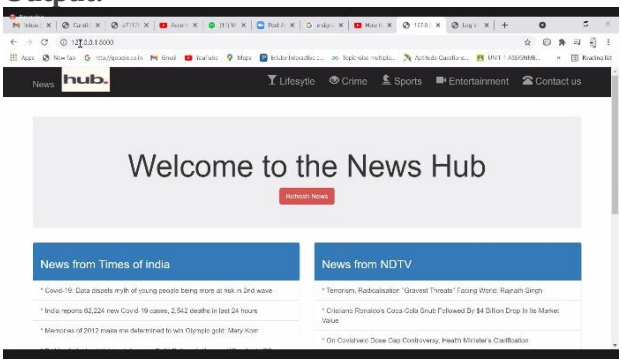

Fig 5. Front page of aggregator

Front page – This is how our front page will look like after scrapping the two websites, Times of India and NDTV. The very first scrapped news comes on the front page as headlines under the name News from Times of India and News from NDTV. We have used the table's concept of bootstrap to showcase these news into a table.

Now, you can configure this to gather your favourite article websites. Although, be wary of blocks. So, web scraping comes at its own cost.

We now know some very cool basics. We also have made a very interesting project to showcase.

III. SUMMARY

Now we have successfully completed the first project in Django. We have used web scraping and Django and python. This integration is as easy as invoking a function in Python.

You can make some more projects in Django using the same concepts and scrap things you want in your project. Django lets you integrate machine learning too.

IV. RESULTS

The application that resulted offers users a full range of options that implement the features we documented so far. The interface (see Figure 3) uses some elements of Material Design [29], a visual language used by Google for their web applications. Once the user has created an account and successfully logged in, he/she is greeted by the main page which contains a list of articles from predefined sources sorted chronologically from the most recent to the oldest.

The list of items is loaded so that only 15 articles are available at the top of the page and as the user scrolls down on page the list of items is expanded.

The user can also use the floating search bar which is always present on top of the page in order to filter articles by keywords. An article is defined by a card element from W3CSS framework. Each article contains a reference image if it was successfully extracted by the download script or otherwise a predefined image, the title of the article, a brief description of the article section, the date of release, and a switch which can be utilizes to add the item to bookmarks for later reading. If the user click on the headline of the section, he will be directly go to the full page, where he/she will get more information about it. To access more site features, we can use the switch in the top of the left corner, that show a sidebar with more assortment choices. Between these we discriminate: the RSS directory, RSS feed management and we the choice to maintain the keywords for Twitter post crawling. Main page with settings navbar expanded.

V. CONCLUSIONS AND FUTURE WORK

The aggregator obtained, as the subject of this paper, is a web application that indexes RSS and News API feeds from different sources such as news platforms, blogs, or online magazines. At the same time, it also provides the ability to index posts from Twitter. The application offers features specific to existing news aggregators, such as choosing sources, grouping them by different criteria, bringing information into a common, easy- to-read format or saving articles for later reading. At the same time, the application corrects some drawbacks such as user privacy or large amounts of ads or spam.

From the application architecture point of view, the aggregator is developed using modern programming languages and technologies that provide scalability and allow further development. In addition to these aspects, the architecture allows it to run on a multi-server cluster, each with a well established role (downloading, storage etc.).

REFERENCES :

1) C. Grozea, D. Cercel, C. Onose and S. Trausan- Matu, "Atlas: News aggregation service," *2017 16th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 2017, pp. 1- 6, doi: 10.1109/ROEDUNET.2017.8123756.
2) M. Mohirta, A. S. Cernian, D. Carstoiu, A. M. Vladu, A. Olteanu and V. Sgarciu, "A semantic Web based scientific news aggregator," *2011 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 2011, pp. 285-289, doi: 10.1109/SACI.2011.5873015.
3) M. Aniche *et al.*, "How Modern News Aggregators Help Development Communities Shape and Share Knowledge," *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, 2018, pp. 499-510, doi: 10.1145/3180155.3180180.
4) S. Khan, T. Khan, C. Prasad, A. Khatri and I. Khan, "Intelligent News Aggregator and Validator," *2019 International Conference on Nascent Technologies in Engineering (ICNTE)*, 2019, pp. 1-5, doi: 10.1109/ICNTE44896.2019.8945945.
5) R. E. Radu, O. Grigorescu and R. V. Rughiniș, "Security News Aggregator," *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 2019, pp. 1-8, doi: 10.1109/ROEDUNET.2019.8909609.
6) K. Weiying, D. N. Pham, N. C. Hai and H. H. Ong, "Topic Modelling for Malay News Aggregator," *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, 2018, pp. 1-6, doi: 10.1109/ICACCAF.2018.8776737.
7) O. Oechslein, M. Haim, A. Graefe, T. Hess, H. Brosius and A. Koslow, "The Digitization of News Aggregation: Experimental Evidence on Intention to Use and Willingness to Pay for Personalized News Aggregators," *2015 48th Hawaii International Conference on System Sciences*, 2015, pp. 4181-4190, doi: 10.1109/HICSS.2015.501.
8) F. Hamborg, N. Meuschke and B. Gipp, "Matrix- Based News Aggregation: Exploring Different News Perspectives," *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2017, pp. 1-10, doi: 10.1109/JCDL.2017.7991561.
9) R. Bahana, R. Adinugroho, F. L. Gaol, A. Trisetyarso, B. S. Abbas and W. Suparta, "Web crawler and back-end for news aggregator system (Noox project)," *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, 2017, pp. 56-61, doi: 10.1109/CYBERNETICSCOM.2017.8311684.
10) K. Sundaramoorthy, R. Durga and S. Nagadarshini, "NewsOne — An Aggregation System for News Using Web Scraping Method," *2017 International Conference on Technical Advancements in Computers and Communications (ICTACC)*, 2017, pp. 136-140, doi: 10.1109/ICTACC.2017.43.
11) V. Latypov, E. V. Ehlakov, N. Ivanov, E. F. Smirnov and I. Y. Khramov, "News Aggregator from Telegram Channels Using Thematic Text Analysis," *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, 2021, pp. 2150-2153, doi: 10.1109/ElConRus51938.2021.9396536.